



TITLE:

ファジイ最適化法のアルゴリズム (計算力学の新解法と領域分割法)

AUTHOR(S):

水藤, 寛; 河原田, 秀夫

CITATION:

水藤, 寛 ...[et al]. ファジイ最適化法のアルゴリズム (計算力学の新解法と領域分割法). 数理解析研究所講究録 2000, 1129: 95-107

ISSUE DATE:

2000-02

URL:

<http://hdl.handle.net/2433/63652>

RIGHT:

ファジイ最適化法のアルゴリズム

千葉大学工学部 水藤 寛 (Hiroshi Suito)

千葉大学工学部 河原田 秀夫 (Hideo Kwarada)

1 はじめに

目的関数の、ある極小値を探索する局所的最適化については種々のアルゴリズムが提案されており、特に線形な問題については多くの有用なアルゴリズムが開発されている。非線形最適化問題についても多くの方法が提案されつつあるが、本研究ではなるべく小さな計算コストで極小値を求めるためにファジイ推論の手法を取り入れたアルゴリズムを提案する。

2 アルゴリズム

2.1 勾配法による極小値探索

目的関数 $f(x)$ の極小値を探索する際に、目的関数の勾配 ($\nabla f(x)$) の情報を用い、反復によって極小値を探索するアルゴリズムを総じて勾配法と呼ぶ。勾配法のアルゴリズムは一般的に次のように書かれる。

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} d^{(k)}, \quad (k > 0), \quad (1)$$

$$x^{(0)} = x_0. \quad (2)$$

ここで、 $d^{(k)}$ は探索方向ベクトル、 $\alpha^{(k)}$ はステップ幅とよばれる。また、上添字は反復のステップ番号を示す。 x_0 は反復の初期値である。

2.2 最急降下法

最急降下法 (Steepest Descent Method) では探索ベクトルとして、その場所の勾配ベクトルの負の方向を用いる。

$$d_s^{(k)} = -\nabla f(x^{(k)}) \quad (3)$$

勾配ベクトルの負の方向 $-\nabla f(x)$ は探索方向として一見最適のようであるが、事実はそのようではない。実際には小さな誤差に敏感であり、極小値の近傍で探索ベクトル列が大きく振動する現象が知られている。そのため、多くの場合、実用的な方法とは言い難い。

この点を直感的に理解するために、例として図(1)のような2次元の目的関数 $f(x)$ の最小値を求める問題を考えよう。図からわかるようにこの目的関数は深い谷を持っており、その中の1点に最小値が存在する。

このような地形があったと考えるとA地点に水を流せば、水は流れて、たしかに一番低い所（最小値）にたまるであろう。しかし、有限なステップ幅を使って探索をおこなう数値的最適化では種々の問題が発生する。例えば、次のような状況を考えてみよう。図(1)のA地点（谷底）において $-\nabla f$ は最小値の存在する方向を指している。しかし、A地点からわずかに谷の斜面を登ったところにあるB地点を考えよう。ここでは $-\nabla f$ はA地点における $-\nabla f$ と90度ずれた方向を向いている。さらにA地点から見てB地点とは逆の方向にわずかに斜面を登ったC地点での $-\nabla f$ は反対方向に90度ずれているのである。

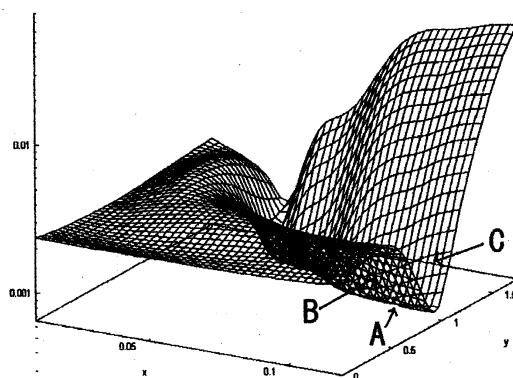


図 1: 目的関数 $f(x)$

図(2)はこの状況を上から見たものである。

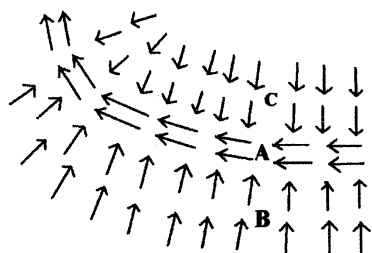


図 2: 谷底近くでの $-\nabla f$

ある探索の過程でB地点にいるとしよう。その地点の勾配の情報に従えばA地点の方向に降りて行くことになる。ちょうどA地点に到達すれば問題はないが、ステップ幅の決定に少しでも誤差があるとA地点で止まらずにC地点まで登って行ってしまふ。そこでの

$-\nabla f$ は今登ってきたのとは逆の方向を向いており、また B 地点の方に向かっていくことになってしまう。つまり、このような地形を各々の地点の勾配の情報のみを使い、しかも有限なステップ幅を用いて数値的に探索しようとするれば、探索ベクトル列がジグザグになるのは避けられないのである。これが前述した探索ベクトル列の振動現象である。

現在提案されている種々の非線型最適化アルゴリズムでは、この現象を回避するような工夫が色々となされている。本研究では少し違った視点からこの現象を回避し、より小さい計算コストで最小値を探索する手法を構築した。

2.3 最急降下法に対する平均化操作の導入

2.3.1 探索ベクトル列の凸結合

前節で述べた最急降下法の問題点を回避するために、探索ベクトル列を何らかの方法で平均化することを考える。

まず、 $d_C^{(k)} = -\nabla f(x^{(k)})$ を求める。次に $d_C^{(k)}$ と $k-1$ 番めの探索ベクトル $d^{(k-1)}$ との凸結合として $d^{(k)}$ を次式のように定義する。

$$d^{(k)} = \beta d^{(k-1)} + (1 - \beta) d_C^{(k)} \quad (4)$$

ここで β は $0 \leq \beta \leq 1/2$ を満たす平均操作係数である。

2.3.2 β の変数の決定

平均操作係数 β の変数を決めるために、目的関数が 2 次元の場合を例にして次のように探索ベクトルの方向についての変動を考える。

$$\begin{aligned} \Delta\theta^{(k)} &= \theta_C^{(k)} - \theta^{(k-1)} \\ \Delta\theta^{(k-1)} &= \theta^{(k-1)} - \theta^{(k-2)} \\ \Delta^2\theta^{(k)} &= \Delta\theta^{(k)} - \Delta\theta^{(k-1)} \end{aligned} \quad (5)$$

ただし、 $\theta_C^{(k)}, \theta^{(k-1)}, \theta^{(k-2)}$ はそれぞれ、 $d_C^{(k)}, d^{(k-1)}, d^{(k-2)}$ と x 軸の正の方向がなす角度である。

$\Delta\theta^{(k)}, \Delta\theta^{(k-1)}$ は探索ベクトルの方向の一次の変動を、 $\Delta^2\theta^{(k)}$ は二次の変動の大きさを表している。

$\Delta\theta^{(k)}$ と $\Delta^2\theta^{(k)}$ を $x_1 = |\Delta\theta^{(k)}|/\pi \in I_1, x_2 = |\Delta^2\theta^{(k)}|/\pi \in I_2$ のように規格化し、これらを平均操作係数の変数として使うことにする。ただし $I_1 = I_2 = [0, 1]$ である。

この x_1, x_2 を変数として、平均操作係数 β を構成することにする。

2.3.3 β の関数形の決定

「探索ベクトルの方向の変化が大きければ探索ベクトル列の平均化をおこなう」という方針のもとで、写像 β ;

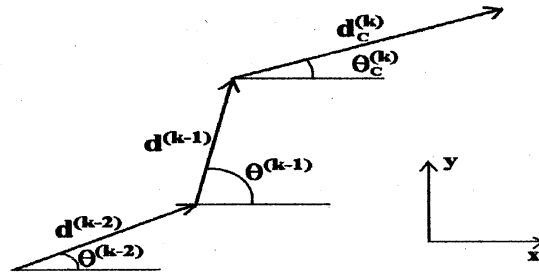


図 3: 探索ベクトルの方向の変化

$$\beta : (x_1, x_2) \in I_1 \times I_2 \mapsto \beta[0, 1/2] \quad (6)$$

には、次のような性質をもたせることにする。

1. β は連続である。
2. $\beta(0, 0) = 0$
3. $\beta(x_1, x_2)$ は x_1 and x_2 それぞれに関して擬単調増加関数である。
4. $\beta(1, 1) = \frac{1}{2}$

2.3.4 C-Aアルゴリズム

前節で挙げたような写像の性質を満たすような簡単な写像として、たとえば

$$\beta(x_1, x_2) = \frac{\max(x_1, x_2)}{2}, \quad (7)$$

ととることができる。このような β を用いて探索ベクトルの平均化操作を行うアルゴリズムをC-A(Averaging)アルゴリズムとよぶことにする。

2.4 決定論的ファジイ平均操作の導入

前節で述べたC-Aアルゴリズムは、探索ベクトル列に対して単に平均化を施すのであるが、この平均化操作にファジイ推論を導入する。

Pre-condition		Post-condition	
x_1	x_2	M_{ij}	
S	S	M_{11}	S
S	M	M_{12}	M
M	S	M_{21}	
M	M	M_{22}	
S	L	M_{13}	L
M	L	M_{23}	
L	S	M_{31}	
L	M	M_{32}	
L	L	M_{33}	

表 1: Fuzzy Ruling Diagram

2.4.1 ファジイ推論ルール

ファジイ推論で用いる Fuzzy ruling diagram を表 1 に示す。表中で S は Small、M は Medium、L は Large をあらわす。例えば入力 x_1, x_2 がともに「小さい」とときには、結論として「小さい」を出力することを示している。

このルールに基づいて推論を行うために、以下のようなファジイ推論の理論で用いられるいくつかの概念を用いる。

2.4.2 三角型ファジイ数の定義

$a_i \in R (i = 1, 2, 3)$, $a_1 < a_2 < a_3$ とする。また、メンバシップ関数 $\mu_A(x)$ を次のように定義する。

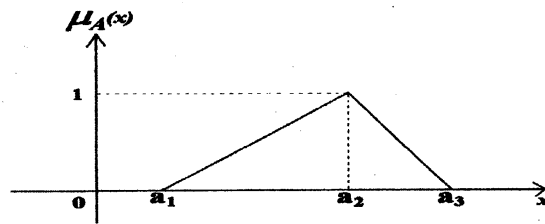
$$\mu_A(x) = \begin{cases} \frac{x - a_1}{a_2 - a_1} & \text{when } a_1 \leq x \leq a_2 \\ \frac{a_3 - x}{a_3 - a_2} & \text{when } a_2 \leq x \leq a_3 \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

このようなメンバシップ関数 $\mu_A(x)$ を持つ集合を三角型ファジイ数 (Triangular Fuzzy Number) A とよび、 $A = (a_1, a_2, a_3)$ と記す。 a_2 は三角型ファジイ数 A の代表値 (mean value) という。(図 (4))

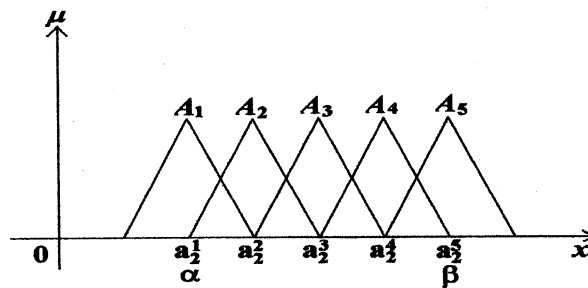
なお、本研究では $a_2 - a_1 = a_3 - a_2$ 、つまり二等辺三角形の三角型ファジイ数を用いた。

2.4.3 ファジイ N 分割の定義

$a < b$, $a, b \in R$ とし、区間 $[\alpha, \beta]$ を $N - 1$ 分割 ($N \geq 3$), $(\alpha = a_1 < a_2 < \dots < a_{N-1} < a_N = \beta)$ して N 個の三角型ファジイ数 $A_i = (a_{i-1}, a_i, a_{i+1})$ ($1 \leq i \leq N$) を定めることを、

図 4: 3 角形ファジイ数 A

区間 $[\alpha, \beta]$ のファジイ N 分割とよぶ。ただし、 $a_0 = \alpha - \frac{\beta - \alpha}{N-1}$, $a_{N+1} = \beta + \frac{\beta - \alpha}{N-1}$ とする。図 (5) は $N = 5$ とした分割の例である。

図 5: ファジイ N 分割 ($N = 5$)

2.4.4 重み関数の定義

ファジイ変数として、 $x = (x_1, x_2) \in R^2$ を選んだ時、 x に対する重み関数 (weight function) w_{ij} を次のように定義する。

$$w_{ij}(x_1, x_2) = \frac{T(\mu_{A_i}(x_1), \mu_{A_j}(x_2))}{\sum_{i,j=1}^N T(\mu_{A_i}(x_1), \mu_{A_j}(x_2))}. \quad (9)$$

ここで、演算 T は次のように定義される。

$$T(a, b) = \min\{a, b\}. \quad (10)$$

この重み関数は明らかに次の性質をもつ。

$$0 \leq w_{ij}(x_1, x_2) \leq 1 \quad (11)$$

$$\sum_{i=1}^N \sum_{j=1}^N w_{ij}(x_1, x_2) = 1 \quad (12)$$

つまり、この w_{ij} は確率密度分布関数の一種とみなすことができる。そして、 $w_{ij}(x_1, x_2)$ の値は x_1 がファジイ数 A_i の代表値 a_i^1 に、 x_2 がファジイ数 A_j の代表値 a_j^2 にどのくらい近いかの物差しになっている。

2.4.5 相関行列の定義

ファジイ変数 x に対して、それらの定義域を I_1, I_2 とする。この時、 I_1, I_2 をそれぞれ N 個の三角型ファジイ数 $A_i = (a_{i1}, a_{i2}, a_{i3}), A_j = (a_{j1}, a_{j2}, a_{j3})$ の代表値 a_{i2}, a_{j2} を要素とする集合 $\{a_{i2}^k\}_{k=1}^N, \{a_{j2}^k\}_{k=1}^N$ で近似する。そして x_1 と x_2 の間の相関をファジイ数 A_i, A_j の代表値 a_{i2}^l, a_{j2}^m ($l, m = 1, 2, \dots, N$) の間の相関で近似する。この相関関係は一つの行列 M_{ij}^N で表すことができる。この行列をファジイ相関行列 (Fuzzy Correlation Matrix) とよぶ。

ファジイ相関行列 M_{ij} は次のような性質を持つことが必要である。

$$M_{ij} \leq M_{i+1,j} \text{ for any } j, \quad (13)$$

$$M_{ij} \leq M_{i,j+1} \text{ for any } i. \quad (14)$$

本研究では、ファジイ相関行列 M_{ij} を次のように定義した。

$$M_{ij} = \frac{\max\{i, j\} - 1}{2(N-1)} \quad (15)$$

2.4.6 単調写像 β^N の定義

重み関数 w_{ij} と相関行列 M_{ij} を用いて、単調写像 β^N を次のように定義する。

$$\beta^N = \sum_{i,j=1}^N w_{ij} M_{ij} \quad (16)$$

ここで、添字の N はファジイ分割数を表す。

このようにして構成された β^N を用いて行う平均操作をファジイ平均とよび、この β^N を用いて平均化操作を行うアルゴリズムを C-FA (Fuzzy Averaging) アルゴリズムとよぶことにする。

2.5 確率論的ファジイ平均操作の導入

前節では決定論的なファジイ平均操作を導入したが、その上に確率的な構造をもたせることを考える。

次のようにして β_N に確率的な構造を持たせる。

1. C-FA method で用いられた $T(a, b) = \min\{a, b\}$ のかわりに、次のように定義された $T_\sigma(a, b)$ を用いる。

(a) $0 \leq a \leq b \leq 1$ とする。区間 $[0, 2\pi]$ を

$$\Omega_a = [0, 2\pi(1 - a^\sigma)], \quad (17)$$

と、

$$\Omega_b = (2\pi(1 - a^\sigma), 2\pi], \quad (18)$$

のふたつの区間に分割する。ただし σ は $1 \leq \sigma \leq +\infty$ の数である。

(b) 乱数 $r \in [0, 2\pi]$ をとる。もし $r \in \Omega_a$ ならば $T_\sigma(a, b) = a$ とし、もし $r \in \Omega_b$ ならば $T_\sigma(a, b) = b$ とする。

2. (9) 式における T を T_σ に置き換えて w_{ij}^σ を構成し、 $w_{ij}^\sigma(x)$ と M_{ij} を用いて (16) 式と同様に $\beta_\sigma^N(x)$ を定義する。

図(6)にこの関係をルーレットに模して示す。図の様に区分けしたルーレットを回して止まった所が Ω_a なら $T_\sigma(a, b) = a$ とし、 Ω_b で止まったならば $T_\sigma(a, b) = b$ とするのである。

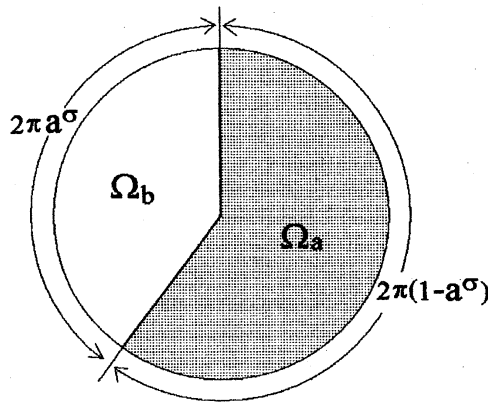


図 6: ルーレット $T_\sigma(a, b)$

このようにある種の確率的構造を持って構成された β_σ^N を用いて平均化操作を行うアルゴリズムを C-SFA (Stochastic Fuzzy Averaging) アルゴリズムとよぶことにする。

2.6 β , β^N , β_σ^N の関係

ここで、 $N \rightarrow \infty$ としたときの β^N , β_σ^N の振る舞いを調べる。領域 $D = [0, 1] \times [0, 1]$ における点 (x_1, x_2) をファジイ数の組 A_i, A_j に対応づけるものとする。

ここで A_i は x_1 軸上をファジイ N 分割したときの i 番目のファジイ数、 A_j は x_2 軸上をファジイ N 分割したときの j 番目のファジイ数をあらわす。

与えられた (x_1, x_2) に対してメンバシップ関数の値が0でないようなファジイ数 A_i, A_j の番号の組 (i, j) が成す集合を $\Lambda_N(x_1, x_2)$ とかく。

$$\Lambda_N(x_1, x_2) = \{(i, j) | (x_1, x_2) \in \text{supp} A_i \times \text{supp} A_j\}. \quad (19)$$

ここで、

$$\forall x_1 \in \text{supp} A_i, \quad \left| \frac{i-1}{N-1} - x_1 \right| \leq \frac{1}{N-1}, \quad (20)$$

$$\forall x_2 \in \text{supp} A_j, \quad \left| \frac{j-1}{N-1} - x_2 \right| \leq \frac{1}{N-1}, \quad (21)$$

が成り立つ。

本研究で採用したように

$$\beta = \frac{\max(x_1, x_2)}{2}, \quad (22)$$

$$M_{ij} = \frac{\max(i, j) - 1}{2(N-1)}, \quad (23)$$

ととった場合を考え、 β と β^N の差を計算してみる。

$$\begin{aligned} \beta^N - \beta &= \sum_{i,j=1}^N w_{ij} M_{ij} - \frac{\max(x_1, x_2)}{2} \\ &= \sum_{i,j=1}^N w_{ij} \left\{ \frac{\max(i, j) - 1}{2(N-1)} - \frac{\max(x_1, x_2)}{2} \right\} \\ &\leq \sum_{(i,j) \in \Lambda_N(x_1, x_2)} \left| \frac{\max(i, j) - 1}{2(N-1)} - \frac{\max(x_1, x_2)}{2} \right| \\ &\leq \frac{1}{2} \sum_{(i,j) \in \Lambda_N(x_1, x_2)} \left| \max\left(\frac{i-1}{N-1}, \frac{j-1}{N-1}\right) - \max(x_1, x_2) \right| \\ &\leq \frac{1}{2} \sum_{(i,j) \in \Lambda_N(x_1, x_2)} \left| \frac{i-1}{N-1} - x_1 \right| + \left| \frac{j-1}{N-1} - x_2 \right| \\ &\leq \sum_{(i,j) \in \Lambda_N(x_1, x_2)} \left| \frac{1}{N-1} \right| \rightarrow 0 \text{ as } N \rightarrow \infty. \end{aligned} \quad (24)$$

ただし、重み関数 w_{ij} は (11)(12) の関係を満たすことを用い、また、(20)(21) の関係を使っている。つまり、(11)(12) の性質を持った重み関数を使って構成した β^N は $N \rightarrow \infty$ の極限で $\beta = \frac{\max(x_1, x_2)}{2}$ に収束する。ここでは重み関数 w_{ij} の性質として (11)(12) の関係のみを使っているので、 w_{ij} の構成に確率的な要素が入るかどうかは関係がない。ゆえに、 β_σ^N もまた $N \rightarrow \infty$ の極限で $\beta = \frac{\max(x_1, x_2)}{2}$ に収束する。

次に、 $\sigma \rightarrow \infty$ としたときの β_σ^N の振る舞いを調べる。(17),(18) より、 $\sigma \rightarrow \infty$ の時は、

$$\Omega_a = [0, 2\pi], \quad (25)$$

となるので、常に $T_\sigma(a, b) = a$ となる。つまり、 β_σ^N は $\sigma \rightarrow \infty$ の極限で β^N に収束することがわかる。

これらの関係を次にまとめておく。

$$\lim_{N \rightarrow \infty} \beta_N = \beta, \quad (26)$$

$$\lim_{N \rightarrow \infty} \beta_N^\sigma = \beta, \quad (27)$$

$$\lim_{\sigma \rightarrow \infty} \beta_N^\sigma = \beta_N. \quad (28)$$

つまり C-FA アルゴリズムはファジィ分割の分割数を無限大にした極限で C-A アルゴリズムに一致し、C-SFA アルゴリズムは σ を無限大にした極限で C-FA アルゴリズムに一致する。

2.7 数値実験

ファジィ最適化法の収束過程を検証するため行うため、例として次のような2変数関数を選んだ。

$$z = f(x, y), \quad (x, y) \in D, \quad D = (0, 1) \times (0, 1)$$

$$f(x, y) = \frac{-1}{150} \left(\frac{1}{0.01 + \left(\frac{\sqrt{3}}{2}x - \frac{y - 0.2}{3} - 0.3 \right)^2} + \frac{1}{0.02 + \left(\frac{x}{2} + \frac{\sqrt{3}}{3}(y - 0.2) - 0.4 \right)^2} \right) \quad (29)$$

図(7)にこの関数の鳥瞰図を示す。この関数は $(0.460, 0.495)$ に最小値をもつ。また、ふたつの険しい溝が存在し、その中央付近に最小値がある。

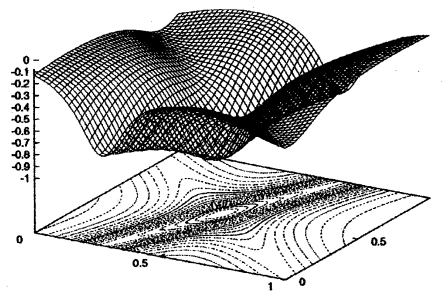


図 7: Cost Function $f(x, y)$

2.7.1 C-A アルゴリズムによる探索ベクトル列の例

図(8)に、C-A アルゴリズムによって探索ベクトル列に平均化を施した例を示す。図(8)で探索ベクトル列がジグザグを描いている付近が、目的関数の中心付近にある深い溝の中にあたっている。

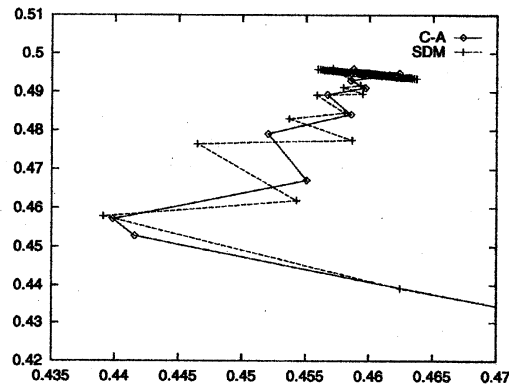


図 8: C-A アルゴリズムによる探索ベクトル列の例

実線がC-A アルゴリズムによる探索ベクトル列、破線が最急降下法による探索ベクトル列である。最急降下法の探索ベクトル列が大きなジグザグを描いているのに対し、C-A アルゴリズムによる平均化を施すと振動が押さえられ、少ない反復回数で最小値の近傍に達しているのが観察される。

2.7.2 C-FA アルゴリズムによる探索ベクトル列の例

図(9)にファジイ平均を用いた場合の探索ベクトル列の例を示す。実線はC-FA アルゴリズムによる探索ベクトル列、破線がC-A アルゴリズムによる探索ベクトル列である。探索対象として選んだ目的関数は前の例と同じである。図(8)より、最小値の近傍を拡大した図である。

最小値の近傍にいくまではC-A アルゴリズムによる探索ベクトル列もC-FA アルゴリズムによる探索ベクトル列も変わりはない。しかし、最小値近傍に近づいてから、C-A アルゴリズムは無駄な振動を続けるのに対し、C-FA アルゴリズムは非常に少ない回数で最小値に収束している。これは、探索ベクトルの方向の変化を粗っぽく評価したために、違う方向から最小値に接近し、そのためにかえってはやく最小値に行き着くことができたのである。この例はファジイ平均操作がよい方向に働いた例であり、別の初期値から探索を開始すればかえって悪くなる結果になることもあり得る。しかし、異なった初期値から探索を開始する実験を繰り返して計算コストの平均をとると、C-FA アルゴリズムの方が、単純な平均化のみをおこなったC-A アルゴリズムより、収束領域の広さ、平均計算コストの両方の点で優れていることがわかるのである。

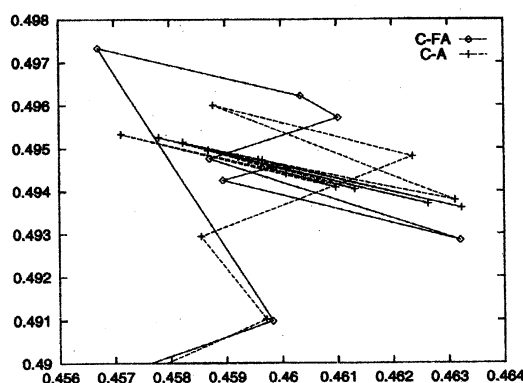


図 9: C-FA アルゴリズムによる探索ベクトル列の例

2.7.3 C-SFA アルゴリズムによる探索ベクトル列の例

図(10)に確率的構造を持ったファジィ平均を用いた場合の探索ベクトル列の例を示す。実線はC-SFA アルゴリズムによる探索ベクトル列、破線がC-FA アルゴリズムによる探索ベクトル列である。探索対象として選んだ目的関数は前の例と同じである。図に示された領域は図(8)の領域をいくらか拡大した領域である。

C-SFA アルゴリズムでは乱数を使用するので、時々平均操作係数 β の値がずれ、探索ベクトルがとんでもない方向に行ってしまう。図(10)はこれがよい方に働き、逆の方向から最小値に近づくことによってずっと早く収束した例である。もちろんこの図のようなことが起きたのは偶然であり、もう一度試行すれば同じ初期値から探索を開始しても違う結果になる。しかし実験を非常に多数回繰り返して計算コストの平均をとると、このアルゴリズムが決定論的なC-FA アルゴリズムに比べて優れていることがわかる。

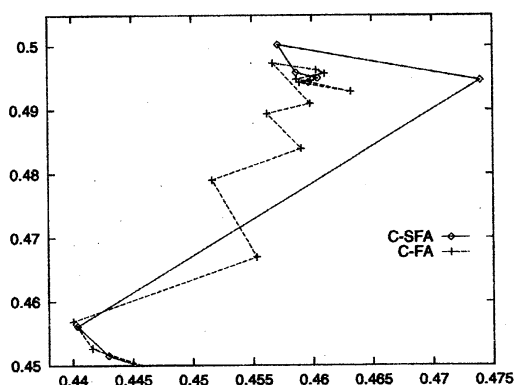


図 10: C-SFA アルゴリズムによる探索ベクトル列の例

3 まとめ

本研究では、ある一つの極小値を探索するアルゴリズムとしてファジイ推論の手法を導入したファジイ最適化法を提案した。また、このアルゴリズムを用いてある目的関数の最小値を求める数値実験を行った。構築した探索アルゴリズムの要点は、

1. 探索ベクトル列の平均化
2. 決定論的ファジイ推論の導入
3. 確率論的ファジイ推論の導入

である。このアルゴリズムの詳細な検証、及び大域的最適化アルゴリズムへの拡張等については、文献 [1][2] を参照されたい。

参考文献

- [1] Kawarada, H. and Suito, H., "Fuzzy Optimization Method", *Computational Science for the 21st Century*, pp 642-651, John Wiley & Sons, 1997.
- [2] Kawarada, H., Ohtomo, T., Periaux, J. and Suito, H., "Multi-Start Fuzzy Optimization Method", *GAKUTO International Series, Mathematical Sciences and Applications*, Vol.11, *Recent Developments in Domain Decomposition Methods and Flow Problems*, 1997.